

УДК 681.3 (075)

UDC 681.3 (075)

**ИЕРАРХИЧЕСКИЕ ОБЪЕКТНЫЕ  
МОДЕЛИ ПРИЛОЖЕНИЙ OFFICE****HIERARCHICAL OBJECT MODELS  
OF OFFICE APPLICATIONS****Ю. А. Быкадоров,**

кандидат физико-математических наук,  
доцент, доцент кафедры информатики  
и методики преподавания информатики;  
Белорусского государственного  
педагогического университета  
имени Максима Танка

ORCID: <https://orcid.org/0009-0002-8729-2271>**Y. Bykadorau,**

candidate of physical and mathematical  
sciences, associate professor, associate  
professor of the Department of Informatics  
and Methods of Teaching Informatics;  
Belarusian State Pedagogical University  
named after Maxim Tank

ORCID: <https://orcid.org/0009-0002-8729-2271>

Поступила в редакцию 13.05.2025.

Received on 13.05.2025.

В статье предложены иерархические объектные модели приложений Word и Excel пакета Office, которые позволяют даже начинающим программистам быстро освоить процесс составления рабочих имен объектов в процедурах на языке VBA. Модель каждого приложения включает несколько графических схем, на которых в иерархическом порядке «родитель – потомок» размещены изображения объектов приложений с их короткими именами и описан порядок составления рабочих имен для основных объектов приложения и их свойств.

Описания основных объектов иерархических моделей сопровождаются простыми примерами команд на языке VBA, в которых рабочие имена объектов построены с помощью иерархических моделей.

В статье используется концепция языка JavaScript, по которой любой объект в процедуре VBA имеет имя и тип, задаваемый другим объектом.

*Ключевые слова:* приложения пакета Office, иерархическая объектная модель, графические схемы, тип объекта, процедура на языке VBA.

The article proposes hierarchical object models of the Word and Excel applications of the Office package, which allow even novice programmers to quickly master the process of composing working names of objects in procedures in the VBA language. Each application model includes several graphical diagrams that display images of application objects with their short names in a parent-child hierarchical order and describe the procedure for composing working names for the main application objects and their properties.

Descriptions of the main objects of hierarchical models are accompanied by simple examples of commands in the VBA language, in which the working names of objects are constructed using hierarchical models.

This article uses the JavaScript concept that any object in a VBA procedure has a name and type that is defined by another object.

*Keywords:* office applications, hierarchical object model, graphical diagrams, object type, VBA procedure.

**Введение.** Объектная модель пакета Office, представленная ее разработчиком на сайте Microsoft Learn [1], включает объектные модели приложений пакета (Word, Excel, PowerPoint, Access и др.) и объектную модель Office, в которой собраны объекты, общие для всех приложений (например, объекты интерфейса приложений). Модели представляют собой списки имен объектов, для которых доступны списки их свойств и методов. Для краткости объектные модели [1] будем называть списочными моделями. Схожие объектные модели приложений Office представлены в [2–3].

Объектная модель пакета Office также представлена разработчиком в редакторе Visual Basic, который встроен в каждое приложение пакета.

Проблема заключается в том, что многие имена объектов, представленные в списочных моделях, не используются в процедурах на языке VBA (Visual Basic for Applications).

Имя объекта в процедуре VBA будем называть рабочим именем объекта.

Каждую часть рабочего имени объекта будем называть коротким именем соответствующего объекта.

Например, чтобы в процедуре VBA для приложения Excel обратиться к диапазону ячеек **A1:B3** на рабочем листе **2** в рабочей книге **Книга1.xlsx**, диапазон как объект языка VBA должен иметь имя

**Application.Workbooks("Книга1.xlsx").Worksheets(2).Range("A1:B3")**

В примере с именем объекта в списочной модели совпадает только короткое имя **Application**, которое можно опустить.

В статье мы будем использовать концепцию языка JavaScript, по которой любой объект в процедуре VBA имеет имя и тип. В соответствии с данной концепцией объект с именем **Workbooks("Книга1.xlsx")** является объектом типа **Workbook**.

Совокупность коротких имен объектов в рабочем имени объекта перед его коротким именем называется квалификатором этого объекта [1].

При создании процедур VBA именно построение квалификатора объекта вызывает основные трудности у новичков. В статье предложены иерархические объектные модели приложений Word и Excel. Эти модели отражают иерархические связи между объектами приложений и помогают составить рабочие имена объектов и их свойств в процедурах VBA.

## 1. Иерархическая объектная модель приложения Word.

### 1.1. Головная схема модели.

Иерархическая объектная модель приложения Word представляет собой набор графических схем. Модель включает головную схему для построения рабочего имени документа и несколько дополнительных схем для построения рабочих имен дочерних объектов документа и их свойств.

Головная схема иерархической объектной модели приложения Word представлена на рисунке 1.



Рисунок 1

На схемах стрелки сверху вниз указывают направление от родительских объектов к дочерним.

Объект **Application** представляет приложение Word и является родительским для всех остальных объектов приложения.

Документы, открытые в приложении, образуют коллекцию (массив) **Documents**, которая является дочерней для объекта **Application**.

Коллекция **Documents** в блоке с закругленными углами содержит как элементы прямоугольники открытых документов (объектов типа **Document**) с короткими именами в виде имен элементов массива. Тип **Document** элементам этой коллекции в процедуре VBA присваивается автоматически.

Открытые в приложении документы также являются дочерними для объекта **Application**.

Открытые документы в процедуре могут иметь и другие короткие имена. В частности, документ, открытый в активном окне приложения, входит в коллекцию **Documents**, но имеет второе короткое имя **ActiveDocument**.

Рабочее имя документа составляется на схеме по стрелке сверху вниз из коротких имен объектов в прямоугольниках, разделяемых точкой. Например,

**Application.Documents(1)**

**Application.ActiveDocument**

Короткое имя коллекции **Documents** не выделено в модели каким-либо образом, поскольку при составлении рабочих имен конкретных объектов играет вспомогательную роль. Хотя использовать это короткое имя в процедурах можно. Вот пример команды VBA.

**MsgBox Application.Documents.Count**

Команда позволяет подсчитать число документов в коллекции **Documents** и вывести результат в диалоговом окне. Свойство **Count** (число объектов в коллекции) дописывается справа к рабочему имени коллекции после точки.

В любых рабочих именах объектов Word короткое имя **Application** можно опускать, если процедура VBA предназначена для работы с объектами приложения Word.

Как продолжение головной схемы иерархической объектной модели предлагаются схемы дочерних текстовых объектов, рисунков и фигур.

### 1.2. Схема дочерних текстовых объектов

Основу документа составляет, конечно, текст, который обычно состоит из текстовых объектов: абзацев, предложений, слов и, наконец, символов.

Для создания рабочего имени текстового объекта документа используется следующая схема, которая дополняет головную схему (рисунок 2).

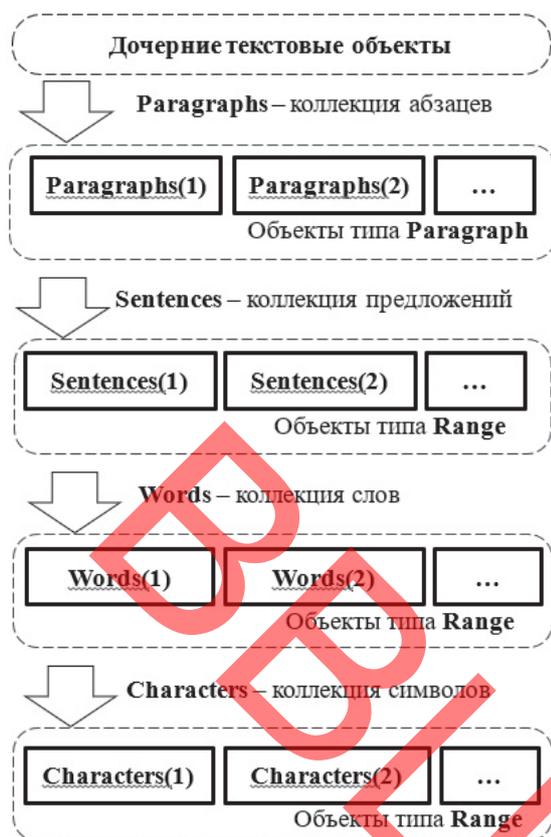


Рисунок 2

Схема дочерних текстовых объектов Word включает четыре коллекции, по числу основных текстовых объектов. На что следует обратить внимание.

Во-первых, представленные коллекции как объекты являются свойствами объекта типа **Document**.

Во-вторых, короткое имя объектов каждой коллекции представлено как имя элемента массива. Элемент коллекции в процедуре может иметь и другие короткие имена. Тип элемента коллекции в процедуре VBA присваивается автоматически.

В-третьих, структура представленной схемы может меняться. Из нее можно исключить любые коллекции, оставив даже только одну. Порядок коллекций должен сохраняться.

В-четвертых, рабочее имя объекта любой коллекции включает рабочее имя документа (из головной схемы) и далее короткие имена предыдущих объектов коллекций (по стрелкам сверху вниз), если таковые используются. Короткие имена разделяются точкой. Например, пятое слово в первом абзаце второго открытого документа будет иметь рабочее имя **Documents(2).Paragraphs(1).Words(5)**

Десятое предложение активного документа имеет рабочее имя

**ActiveDocument.Sentences(10)**

В коллекции **Paragraphs (Абзацы)** элементы имеют тип **Paragraph**. Свойства и методы для объектов типа **Paragraph** можно посмотреть в списочной модели [1]. А вот элементы всех остальных коллекций имеют тип **Range** (Диапазон).

### 1.3. Объекты типа Range

Объект типа **Range** (Диапазон) в приложении Word – это непрерывный фрагмент текста.

Объект типа **Range** может включать весь документ, несколько абзацев, слово и несколько символов. Диапазоны создаются и используются только в процедурах VBA.

Некоторые объекты документа получают тип **Range** автоматически. Этот факт отражен на схеме дочерних текстовых объектов. Отличный от **Range** тип среди текстовых объектов имеют только абзацы (объекты типа **Paragraph**).

Чтобы объектами типа **Paragraph** можно было работать как с объектами типа **Range**, нужно изменить их тип. Для этого используют свойство **Range** этих объектов. Получаются рабочие имена объектов типа **Range**. Например,

**ActiveDocument.Paragraph(2).Range**

Практически получается по содержанию тот же объект, но другого типа.

Теперь к этим объектам можно применять свойства и методы объектов типа **Range**, которых достаточно много в списочной модели.

Следует сказать, что для объекта типа **Paragraph** изменение его типа не отменяет возможность использования для него схемы текстовых дочерних объектов: предложения, слова, символы. Вот пример команды подсчета числа элементов коллекции **Words** в заданном абзаце типа **Range**.

**Msg BoxActiveDocument.Paragraphs(8).Range.Words.Count**

В диалоговое окно команда выведет число слов в 8-м абзаце активного документа.

### 1.4. Объект Selection

Объект **Selection** (Выделение) – это выделенный фрагмент документа. Если в документе ничего не выделено, то объект представляет положение текстового курсора.

Выделенный фрагмент документа выделяется на экране серым цветом фона. В каждом документе может быть только один выделенный фрагмент, т. е. один объект **Selection**.

Выделять фрагмент в документе можно вручную (мышью или клавиатурой) или с помощью команд процедуры VBA.

В процедуре VBA для выделения фрагмента используют метод **Select**, который дописывают после точки к рабочему имени объекта типа **Document** или к рабочему имени объекта типа **Range**. Например,

**ActiveDocument.Select****ActiveDocument.Paragraphs(7).Range.Select**

Рабочее имя объекта **Selection** включает только два коротких имени. Например,

**Documents(2).Selection**

Если же объект **Selection** рассматривается в активном окне документа, которое имеет короткое имя **ActiveWindow**, то его рабочее имя совпадает с коротким именем **Selection**.

Чтобы работать с текстом объекта **Selection**, можно построить гибридную иерархическую модель. Для этого вместо головной схемы надо использовать только объект **Selection**, а схему дочерних текстовых объектов оставить. При этом его дочерние текстовые объекты сохраняют свой тип согласно схеме дочерних текстовых объектов.

**1.5. Дочерние рисунки и фигуры**

Документ *Word* на своих страницах имеет текстовый слой и слой рисования. Рисунки и фигуры могут размещаться в слое рисования, а рисунки – в текстовом слое как большие символы. В процедурах *VBA* используют разные объекты для рисунков и фигур в разных слоях, поэтому в иерархической модели предлагается несколько схем дочерних рисунков и фигур.

Для составления рабочего имени дочернего рисунка или фигуры в слое рисования документа как дополнение головной схемы предлагается следующая схема (рисунок 3).

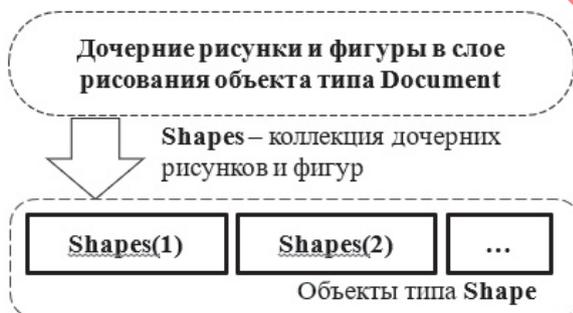


Рисунок 3

Коллекция **Shapes** включает все рисунки и фигуры в слое рисования документа, в том числе рисунки и фигуры в колонтитулах, если они там есть. Рисунки и фигуры являются объектами типа **Shape**. Тип присваивается автоматически.

Рабочее имя рисунка или фигуры в слое рисования включает рабочее имя документа из головной схемы и короткое имя рисунка или фигуры их дополнительной схемы. Эти имена разделяются точкой.

Как дополнение для объектов типа **Shape** представлена третья схема с короткими именами основных свойств этих объектов, причем некоторые из свойств являются объектами (рисунок 4).

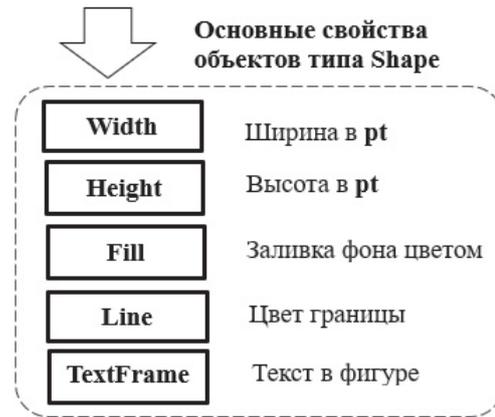


Рисунок 4

Короткое имя свойства дописывается справа к рабочему имени рисунка или фигуры после точки.

Значения ширины и высоты объекта типа **Shape** в процедуре задаются в пунктах (**pt**) без указания единиц измерения. Например, **ActiveDocument.Shapes(1).Height = 50**

Иногда к рабочему имени объекта надо добавлять целую цепочку свойств. Так, цепочкой свойств **ForeColor.RGB** дополняются свойства **Fill** и **Line**, чтобы установить конкретный цвет. Пример команды, которая устанавливает красный цвет фона заданной фигуры:

**Documents(1).Shapes(2).Fill.ForeColor.RGB = vbRed**  
Код цвета в команде задан константой цвета, но можно также использовать функцию **RGB()**, например **RGB(255, 0, 0)** для красного цвета.

Для работы с текстом, который содержит фигура, используют цепочку свойств **TextFrame.TextRange**. Например, команда записи в заданную фигуру текста имеет вид.

**Document(1).Shapes(3).TextFrame.TextRange = «Текст»**

Полный список свойств и методов объекта типа **Shape** можно просмотреть в списочной модели.

А вот коллекция рисунков и фигур в слое рисования объекта **Selection** носит другое имя, и иерархическая объектная модель этого объекта уместается в одну схему (рисунок 5).

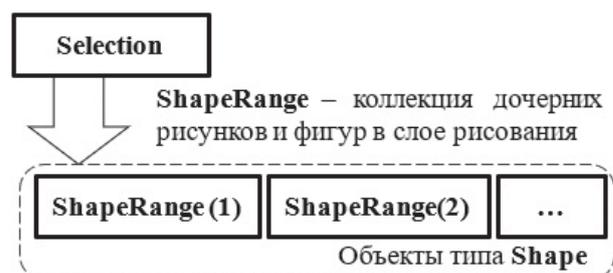


Рисунок 5

Рисунки и фигуры в слое рисования объекта **Selection** являются объектами типа **Shape**, как и элементы коллекции **Shapes** для документа. Поэтому для объекта **Selection** как дополнительную можно использовать схему с набором основных свойств объектов типа **Shape**.

Выделить в слое рисования можно и только одну фигуру. Тогда, например, задать ей зеленый цвет контура позволяет команда

```
Selection.ShapeRange(1).Line.ForeColor.RGB = vbGreen
```

Для создания рабочих имен дочерних рисунков в текстовом слое документа предназначена другая схема, которая добавляется к основной (рисунок 6).

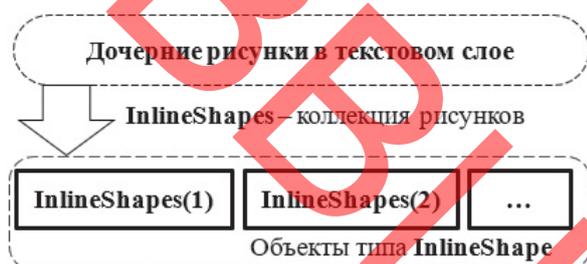


Рисунок 6

Рабочие имена дочерних рисунков в текстовом слое также строятся добавлением короткого имени рисунка к рабочему имени документа (справа после точки).

Набор свойств объекта типа **InlineShape** поменьше набора свойств объекта типа **Shape**. Естественно, список свойств объекта типа **InlineShape** представлен в списочной модели.

Объект **Selection** может содержать рисунки в текстовом слое. Для создания рабочего имени таких рисунков надо головную схему заменить одним объектом **Selection**. А как дополнительную использовать схему дочерних рисунков в текстовом слое.

Текстовый объект типа **Range** также может содержать рисунки в текстовом слое. Для создания рабочего имени рисунка в таком объекте следует сначала построить рабочее имя текстового объекта типа **Range**, а затем добавить к нему короткое имя дочернего рисунка с использованием дочерней схемы для рисунков в текстовом слое.

## 2. Иерархическая объектная модель приложения Excel

### 2.1. Головная схема модели

Документ приложения Excel называется книгой. Книга включает несколько рабочих листов. Рабочий лист представляет собой таблицу с прямоугольными ячейками. Целью

процедур VBA в приложении Excel обычно является изменение данных в ячейках рабочего листа, формы ячеек и оформления рабочего листа. Иерархическая объектная модель приложения Excel включает несколько схем.

Головная схема иерархической объектной модели приложения Excel представлена на рисунке 7.

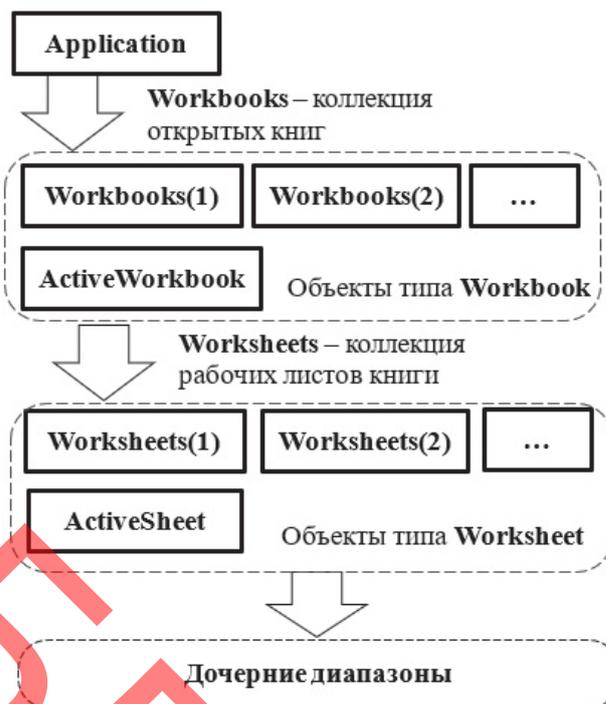


Рисунок 7

Головная схема позволяет создавать рабочее имя рабочего листа Excel.

Иерархия объектов приложения Excel на схеме понятна после знакомства с иерархией объектов приложения Word.

Открытые книги коллекции **Workbooks** в процедуре VBA могут иметь и другие короткие имена. В частности, книга, рабочий лист которой виден в открытом окне **ActiveWindow** приложения, имеет второе короткое имя **ActiveWorkbook**.

Рабочие листы коллекции **Worksheets** также могут иметь и другие короткие имена. В частности, рабочий лист в открытом окне **ActiveWindow** приложения имеет второе короткое имя **ActiveSheet**.

Рабочее имя листа составляется на схеме по стрелке сверху вниз из коротких имен объектов в прямоугольниках и включает три коротких имени, разделенных точками. Например,

```
Application.Workbook(2).Worksheet(4)
Application.ActiveWorkbook.Worksheet(3)
```

В любых рабочих именах объектов Excel короткое имя **Application** можно опустить, если процедура VBA предназначена для работы с объектами приложения Excel. Рабочее имя листа таким образом сокращается.

Если же рабочий лист открыт в окне **ActiveWindow** приложения, то для такого листа в процедуре VBA можно использовать только короткое имя.

**ActiveSheet**

### 2.2. Дочерние диапазоны

Объект типа **Range** (Диапазон) в приложении Excel – это диапазон смежных ячеек на рабочем листе, который обычно имеет вид прямоугольника.

Диапазоны создаются и используются только в процедурах VBA.

В первую очередь в процедурах VBA на рабочем листе создаются объекты типа **Range** для диапазонов, которые имеют имена приложения Excel. Поэтому дополнительная схема для дочерних диапазонов рабочего листа имеет следующий вид (рисунок 8).

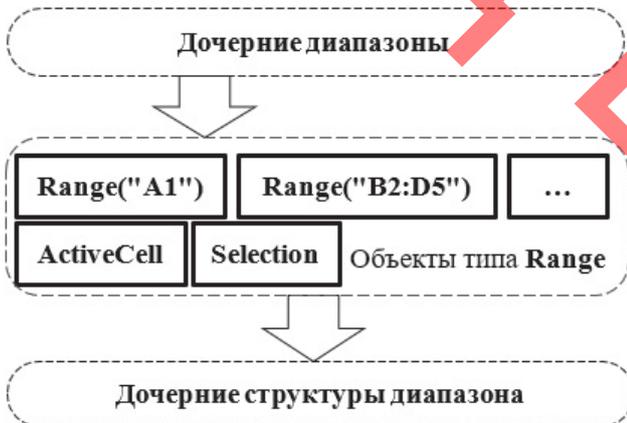


Рисунок 8

Короткие имена объектов типа **Range** включают имена диапазонов приложения Excel. Рабочее имя диапазона в процедуре VBA включает рабочее имя листа и короткое имя диапазона в процедуре VBA (правее после точки). Например,

**ActiveSheet.Range("B3")**

**ActiveSheet.Range("C2:E5")**

На любом рабочем листе ячейка, на которую указывает табличный курсор, является объектом типа **Range** и имеет короткое имя **ActiveCell**. На рабочем листе, открытом в окне **ActiveWindow** приложения, короткое имя **ActiveCell** становится рабочим именем

активной ячейки (без указания квалификатора). Например, следующая команда VBA выводит в диалоговое окно имя активной ячейки на рабочем листе, открытом в окне **ActiveWindow** приложения.

**MsgBox ActiveCell.Address**

Объект **Selection** (Выделение) – это выделенный диапазон ячеек рабочего листа. Если на рабочем листе диапазон не выделен, выделенным диапазоном считается активная ячейка. Объект **Selection** имеет тип **Range**. Выделенный диапазон выделяется на экране серым цветом фона его ячеек.

Выделить диапазон ячеек можно вручную (с помощью мыши или клавиатуры) и командами процедуры VBA.

Чтобы в процедуре VBA выделить диапазон, надо к его рабочему имени дописать имя метода **Select** (справа после точки). Например,

**ActiveSheet.Range("C2:E5").Select**

Если на рабочем листе выделен диапазон, то активная ячейка на этом листе все равно существует. Обычно это левая верхняя угловая ячейка выделенного диапазона.

### 2.3. Дочерние структуры диапазона

Любой объект типа **Range** (Диапазон) в приложении Excel имеет свои строки, столбцы и свою коллекцию ячеек. Это и есть дочерние структуры объекта типа **Range**.

Для создания полных имен структурных элементов диапазона предлагается дополнительная схема дочерних структур диапазона (рисунок 9).

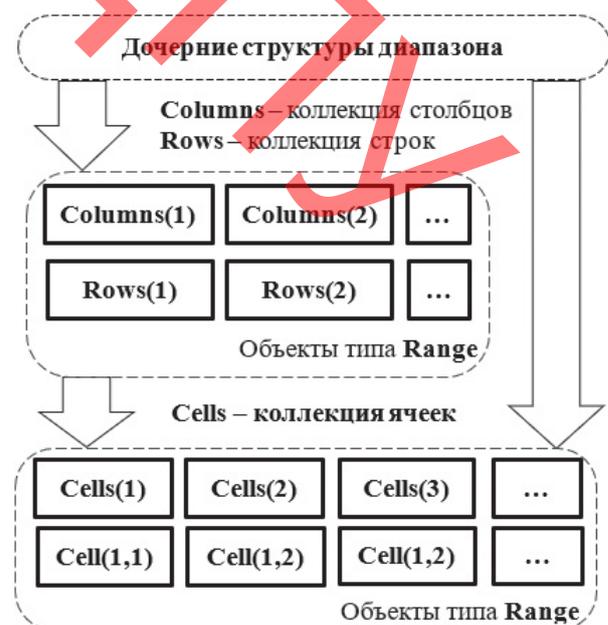


Рисунок 9

Коллекция **Cells** создается для ячеек диапазона, для ячеек его строки или столбца. Короткое имя ячейки коллекции вида **Cell(i)**, включает *i* – последовательный номер ячейки в коллекции. При этом сначала номера получают ячейки первой строки диапазона, затем второй и т. д.

Ячейки диапазона могут использовать и другие короткие имена вида **Cell(i, j)**, где используется номер строки и номер столбца диапазона.

Все перечисленные дочерние структуры можно использовать также для ячеек рабочего листа. Например, чтобы выделить все ячейки рабочего листа, открытого в окне приложения, можно для коллекции ячеек листа использовать команду

**Cells.Select**

#### ЛИТЕРАТУРА

1. Справочник по VBA для Office // Learn. – URL: <https://learn.microsoft.com/ru-ru/office/vba/api/overview/> (дата обращения: 10.05.2025).
2. Object model of the Excel VBA classes // CODE. – URL: <https://www.codevba.com/excel/index.html> (дата обращения: 10.05.2025).
3. Index page of the Word VBA classes // CODE. – URL: <https://www.codevba.com/word/index.html> (дата обращения: 10.05.2025).

Объект **Selection** также является объектом типа **Range**. Поэтому его иерархическая модель включает сам объект, как головной, и дополнительную схему дочерних структур диапазона.

**Заключение.** В статье представлены иерархические объектные модели двух приложений Office. Предложенные модели будут полезны для начинающих изучение языка VBA с целью создания процедур на языке VBA для приложений Word и Excel и могут быть включены в дидактические материалы к лабораторным работам для студентов физико-математических факультетов, изучающих способы автоматизации работы в офисных приложениях.

#### REFERENCES

1. Spravochnik po VBA dlya Office // Learn. – URL: <https://learn.microsoft.com/ru-ru/office/vba/api/overview/> (data obrashcheniya: 10.05.2025).
2. Object model of the Excel VBA classes // CODE. – URL: <https://www.codevba.com/excel/index.html> (data obrashcheniya: 10.05.2025).
3. Index page of the Word VBA classes // CODE. – URL: <https://www.codevba.com/word/index.html> (data obrashcheniya: 10.05.2025).